Описание.

Applepen - это большая торговая сеть, которая занимается продажей всего двух продуктов: яблок и карандашей. Ее магазины расположены в различных уголках Соединенных Штатов и более 10 лет обслуживают покупателей. Недавно топ-менеджмент компании решил более активно использовать имеющиеся у них данные в принятии решений. Каждый магазин собирает информацию о:

1. закупках (поставки яблок и карандашей два раза в месяц)
2. продажах (лог транзакций, по записи на каждую проданную позицию)
3. инвентарь (месячные данные общего количества яблок и карандашей на складе).

Данные доступны в формате CSV. Внутри файла данные отсортированы по дате. Постановка задачи Нам необходимо составить по этим данным три новые таблицы: 1. состояние склада на каждый день 2. месячные данные о количестве сворованного товара 3. агрегированные данные об объемах продаж и количестве сворованной продукции по штату и году.

0. Подготовка данных.

Используем данные, доступные по ссылке:
https://console.cloud.google.com/storage/browser/artem-pyanykh-cmc-prac-task3-seed17/out/input/

Для работы с Google Cloud используем пакет консольных инструментов gcloud. Устанавливаем gsutil как часть Google Cloud SDK.
https://cloud.google.com/storage/docs/gsutil_install

1. Состояние склада на каждый день

Функция day_sell.

Создаем таблицу со столбцом “apple or pen”

Из выборки sell преобразуем данные о транзакциях в таблицу с значениями ap или p.

sell['sku_num'].apply(lambda x: 'apple' if x.find('ap') == 6 else 'pen'): в колонке *'sku_num'* функция *apply* – делает замену всей транзакции на товар, который был куплен по этой транзакции. *Lambda x* – объявление анонимной функции, зависящей от x. x.find('ap') == 6 – сравнение на 6 символе.

newdf = pd.crosstab(df.index, df['apple or pen']): создаем новую таблицу, в которой будет два столбца ”apple” и “pen”. В строках количество за каждый день, значения группируются по индексу(дате).

Присваиваем отрицательные значения, так как это продажа.

Функция day_store.

Объединяем таблицы поставок и продаж.

df = pd.concat([supply, d_sell]).sort_index(). Sort_index() – сортирует по индексам (по возрастанию), *concat* – присоединяет одну таблицу к другой.

df = df.resample('D').sum() - Группирует по дням и суммирует (если в день была и поставка и продажа).

df['apple'] = df['apple'].rolling(str(df.index.size) + 'D').sum(). – считает сумму за месяц.

Rolling – скользящее суммирование, где *str(df.index.size) + 'D'* – ширина окна по дням, по которому производится суммирование.

2. Месячные данные о количестве сворованного товара

Функция stolen

Дано:

supply – исходные данные

inventory - исходные данные

d_sell – данные о проданных товарах.

Создаем новые таблицы с данными о продажах и кражах, сгруппированных по месяцам.

Создаем DataFrame stole, в котором будем собирать результат. В него помещаем количество закупленных товаров + (-1) количество проданных товаров. Это предполагаемое количество, которое должно остаться в конце месяца

В DataFrame res считаем, сколько реально осталось, вычитая из остатка i-го месяца остаток i+1-го месяца. Отдельно считаем нулевой месяц.

Вычитая из предполагаемого остатка реальный остаток, получаем количество сворованных товаров.

3. **Агрегированные данные об объемах продаж и количестве сворованной продукции по штату и году**

Дано:

steal – двумерный массив из DataFrame, в каждом из которых хранятся данные об украденных товарах в конкретном магазине конкретного штата по месяцам

d_s – двумерный массив из DataFrame, в каждом из которых хранятся данные о проданных товарах в конкретном магазине конкретного штата по дням

Создаем три вспомогательных массива, каждый из которых состоит из трех Dataframe. Один DataFrame содержит данные об одном штате.

agr_stolen – в нем будут храниться данные об украденных товарах

agr_sold – в нем будут храниться данные о проданных товарах

agr_all – в нет будет храниться объединенная инфрмация

Далее, все операции будут делать в цикле, для каждого штата

В массиве agr_stolen создаем для штата DataFrame со столбцами ‘apple’ и ‘pen’

В массиве agr_sold создаем для штата DataFrame со столбцами ‘apple’ и ‘pen’

В цикле по количеству магазинов в штате, объединяем таблицы с информацией о проданных/сворованных товарах в DataFrame с помощью метода pandas.concat(). В этой же строке группируем данные по составляющей года от индекса, которым является дата.

Переименовываем столбцы таблиц.

Объединяем таблицы с данными о проданных и сворованных товарах, с помощью метода pandas.merge().

Добавляем в таблицу дополнительный столбец с информацией о штате. Названия штатов предварительно помещаем в объект pandas.Series.

Объединяем данные по всем штатам в одну таблицу и сортируем по году и штату.

Для сверки результатов была написана функция check.